

# Package ‘gridGraphics’

December 13, 2020

**Title** Redraw Base Graphics Using 'grid' Graphics

**Version** 0.5-1

**Description** Functions to convert a page of plots drawn with the 'graphics' package into identical output drawn with the 'grid' package. The result looks like the original 'graphics'-based plot, but consists of 'grid' grobs and viewports that can then be manipulated with 'grid' functions (e.g., edit grobs and revisit viewports).

**Depends** grid, graphics

**Imports** grDevices

**Suggests** magick (>= 1.3), pdftools (>= 1.6)

**License** GPL (>= 2)

**URL** <https://github.com/pmur002/gridgraphics>

**NeedsCompilation** no

**Author** Paul Murrell [cre, aut],  
Zhijian Wen [aut]

**Maintainer** Paul Murrell <paul@stat.auckland.ac.nz>

**Repository** CRAN

**Date/Publication** 2020-12-13 21:20:03 UTC

## R topics documented:

grid.echo . . . . .	2
plotdiff . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

`grid.echo`*Echo **graphics** output using **grid graphics***

---

### Description

Convert a scene that was drawn using the **graphics** package to an identical scene drawn with the **grid** package.

### Usage

```
grid.echo(x = NULL, newpage = TRUE, prefix = NULL, device = offscreen)
```

```
echoGrob(x = NULL, prefix = NULL, device = offscreen, name = NULL)
```

### Arguments

<code>x</code>	Either NULL, a recorded plot, or a function. If NULL, the current graphics device is echoed.
<code>newpage</code>	A logical value indicating whether to start a new page. If FALSE, echoing occurs in the current <b>grid</b> viewport.
<code>prefix</code>	A character value used as a prefix for all grob and viewport names. The default prefix is "graphics".
<code>device</code>	A function that opens a graphics device for <code>grid.echo()</code> to work on. By default this is an off-screen, in-memory device based on the pdf device. This default device may not be satisfactory when using custom fonts.
<code>name</code>	A character identifier.

### Details

If the first argument is a function, it must be a function with zero arguments. If the function needs access to non-global data, use a closure. The function should not call functions that create or destroy graphics devices, or change the current graphics device.

### Value

The `echoGrob` function returns a grob that will echo `x` when it is drawn.

The `grid.echo` function is called for its side-effect of drawing on the current graphics device.

### Author(s)

Paul Murrell

**Examples**

```
## Not run:
# Echo existing drawing
plot(1)
grid.echo()

# Echo result of call to a plotting function
plotfun <- function() plot(1:10)
grid.echo(plotfun)

# Echo result of a plotting function (anonymous) into current viewport
grid.newpage()
pushViewport(viewport(x=0, width=.5, just="left"))
grid.rect(gp=gpar(col=NA, fill="grey"))
grid.echo(function() plot(1:10), newpage=FALSE)

## End(Not run)
```

---

plotdiff

*Test for identical output from grid.echo()*


---

**Description**

Functions to generate a scene using the **graphics** package, reproduce the scene using `grid.echo()`, test whether the two results are identical, and report on any differences.

**Usage**

```
plotdiff(expr, label, dev = "pdf",
         antialias = TRUE, density = 100, width = 7, height = 7)
plotdiffInit()
plotdiffResult(warn = FALSE)
```

**Arguments**

<code>expr</code>	An expression that draws something using the <b>graphics</b> package.
<code>label</code>	A character value that is used to label files generated during testing.
<code>dev</code>	The graphics device used for drawing and echoing. Currently can only be either "pdf" or "png".
<code>antialias</code>	A logical value indicating whether to perform antialiasing when converting from PDF to PNG.
<code>density</code>	A numeric value indicating the resolution (dpi) to use when converting from PDF to PNG.
<code>width, height</code>	Numeric values indicating the size of the device to test on.
<code>warn</code>	A logical value indicating whether non-identical output should produce a warning or an error.

## Details

In default usage, `plotdiff()` is used to generate two PDF files, one using the original expression and the other from a call to `grid.echo()`. The PDF files are then converted to PNG files and the PNG files are compared (using ImageMagick). If there are any differences, the comparison generates a further PNG file that shows the differences.

All files are currently generated in the current working directory.

Text messages are also generated by `plotdiff()` whenever a difference is found, but those messages are stored up rather than printed immediately. The `plotdiffResult()` function prints out all messages since the last call to `plotdiffInit()`.

Standard usage involves calling `plotdiffInit()`, followed by one or more `plotdiff()` calls, then finally a call to `plotdiffResult()`. Examples of the usage of these functions are provided in the numerous test scripts in the `test-scripts` directory of the package.

## Value

All functions are run for their side effects. In the case of `plotdiff()`, the generation of PDF and PNG files and the accumulation of differences about messages. In the case of `plotdiffResult()`, a print out of the accumulated messages, plus possibly either an error or warning.

## Author(s)

Paul Murrell

## See Also

[grid.echo](#)

## Examples

```
## Not run:  
plotdiff(expression(plot(1)), "plot")  
  
## End(Not run)
```

# Index

## \* **dplot**

grid.echo, [2](#)

plotdiff, [3](#)

echoGrob (grid.echo), [2](#)

grid.echo, [2](#), [4](#)

plotdiff, [3](#)

plotdiffInit (plotdiff), [3](#)

plotdiffResult (plotdiff), [3](#)